PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:	10/646,289
Filing Date:	August 21, 2003
Applicant:	Son Ho et al.
Group Art Unit:	2188
Examiner:	Kaushikkumar M. Patel
Title:	LINE CACHE CONTROLLER
Attorney Docket:	MP0390.I

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

PRE-APPEAL BRIEF REQUEST FOR REVIEW AND PETITION FOR EXTENSION OF TIME

Applicants request a Pre-Appeal Brief Conference and contend that the combination of Zaidi and Loafman fails to teach or suggest the elements of the presently pending claims.

Applicants hereby petition under the provisions of 37 C.F.R. § 1.136(a) for an extension of time in which to respond to the outstanding Office Action and includes a fee as set forth in 37 C.F.R. § 1.17(a) with this response for such extension of time.

STATUS OF CLAIMS

Claims 1-2, 5-7, 10, 13-15, 18-19 and 25 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Zaidi et al. (U.S. Pat. No. 6,601,126 B1), Jeddeloh (U.S. Pat. No. 7,133,972 B2) and Loafman (U.S. Pub. No. 2005/0021916 A1. Claims 4, 9, 12, 17, 26 and 33 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Zaidi et al. (U.S. Pat. No. 6,601,126 B1), Jeddeloh (U.S. Pat. No. 7,133,972 B2) and Loafman (U.S. Pub. No. 2005/0021916 A1) as applied to claims 1-2, 5-7 and 13-15 above, and further in view of Barroso et al. (U.S. Pat. No. 6,725,334 B2).

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a memory storage system that is accessed by a first central processing unit (CPU). The memory storage system includes a line cache including a plurality of pages that are accessed by the first CPU. A memory stores data that is loaded into the line cache when a miss occurs. **When the miss occurs and *before a second miss* occurs**, n pages of the line cache are loaded with data from sequential locations in the memory, wherein n is greater than one.

Independent claims 5, 10, 13, 18, and 26 recite similar subject matter. For example, claims 5, 13, 18, and 26 recite preventing **any additional misses after an initial miss**. Claim 10 recites loading data from sequential locations before a second miss occurs.

ARGUMENT

With respect to claim 1, Applicants respectfully submit that Zaidi, either singly or in combination with Jeddeloh and Loafman, fail to at least show, teach, or suggest a line cache including a plurality of pages that are accessed by the first CPU and a first memory device that stores data that is loaded into said line cache when a miss occurs, wherein when said miss occurs and before a second miss occurs, n pages of said line cache are loaded with data from sequential locations in said first memory device, wherein n is greater than one. Loafman appears to disclose that at least two misses occur before prefetching data from multiple sequential memory locations.

The present claims are directed to loading the cache with data from sequential memory locations after a first cache miss and before a second cache miss. (See Page 17 of the Response filed March 5, 2007; hereinafter "the Response"). When the cache miss occurs, n (wherein n is greater than one) pages of line cache are loaded with data in sequential memory locations from a selected memory. In other words, the n pages of the line cache are loaded with the data from the sequential locations **in response to an initial cache miss and before a second (i.e. any additional) cache miss**.

The Examiner acknowledges that Zaidi fails to disclose this limitation and instead relies on Paragraph [0026] of Loafman. (See Page 5, Line 21, through Page 6, Line 1 of the Office Action mailed May 7, 2007; hereinafter "the present Action"). Further, during a telephonic interview conducted on March 5, 2007, the Examiner noted that Loafman discloses that pre-fetching is known in the art. In the Response, Applicants noted that the cited portion discloses retrieving the consecutive pages after observing **a pattern of successive page accesses**. As such, **at least two misses are required**.

For example, Paragraph [0012] recites that "pre-fetching works splendidly when data is being read sequentially" and notes that "after two consecutive page faults of sequentially stored data, a block of sequential pages of data will be pre-fetched." In other words, Applicants respectfully note that Loafman still discloses that the pre-fetching is only performed after at least two consecutive misses (i.e. page faults). (See Pages 17-19 of the Response).

In the present Action, the Examiner appears to acknowledge that Loafman discloses prefetching sequentially after at least two page faults (i.e. misses). (See Page 6, Lines 16-22 of the present Action), which state:

> It is apparently clear from the above statements that Loafman is not prefetching pages if data [is] being read randomly, but since [the] program requests two sequential pages (even though program accesses data randomly, it is not totally random and can access pages sequentially), two page faults will be raised.

In other words, the Examiner's interpretation appears to be analogous to Applicants' position that Loafman discloses that **at least two cache misses are required** before any sequential prefetching. The Examiner further notes at Page 7, Lines 15-16 of the present Action:

> Thus, it is apparently clear that Loafman uses two page faults (two cache misses) to two sequential pages to confirm that data is being read sequentially.

As such, Applicants respectfully submit that the combination of Zaidi and Loafman appear to be absent of any teaching or suggestion of loading multiple pages of the line cache from sequential memory locations in response to a cache miss and before a second cache miss. Instead, the Examiner alleges that it would be obvious to modify Loafman in this manner "if the program reads data sequentially...since it is known that data is being read sequentially." (See Page 7, Lines 17-22 of the present Action)

Applicants respectfully note that none of the references, in particular Loafman, teach or suggest any "prior knowledge" of sequential data reading that the Examiner appears to suggest. In contrast, Loafman clearly discloses that at least two cache misses are required to determine whether a program reads data sequentially. In other words, according to Loafman, a determination that a program is reading data sequentially requires at least two consecutive cache misses. **The Examiner fails to provide any reference that teaches or suggests any knowledge of sequential data reading that is not based on at least two cache misses**.

Applicants respectfully submit that the combination of Zaidi and Loafman appear to be absent of any teaching or suggestion of loading multiple pages of the line cache from sequential memory locations in response to a cache miss and before a second cache miss. Accordingly, Applicants respectfully submit that the presently pending claims are in condition for allowance.

Respectfully submitted,

Dated: September 5, 2007

By:__________________________
Michael D. Wiggins, Reg. No. 34,764
Damian M. Aquino, Reg. No. 54,964

HARNESS, DICKEY & PIERCE, P.L.C.
P.O. Box 828
Bloomfield Hills, Michigan 48303
(248) 641-1600

MDW/dma